Providing Evidence for the Validity of the Virtual Verification of Automated Driving Systems

Birte Neurohr[1 [0000−0002−9982−0206]], Thies de Graaff[1 [0009−0006−5918−9524]], Andreas Eggers[2], Tom Bienmüller[2], and Eike Möhlmann[1 [0000−0003−3815−6353]]

[1] German Aerospace Center (DLR), Escherweg 2, 26121 Oldenburg, Germany
{Birte.Neurohr, Thies.Degraaff, Eike.Moehlmann}@dlr.de

[2] BTC Embedded Systems AG, Gerhard-Stalling-Straße 19, 26135 Oldenburg, Germany
{Andreas.Eggers, Tom.Bienmueller}@btc-embedded.com

Abstract. With the increasing complexity of automated driving systems, formal verification as well as statistical verification that solely relies on real-world testing methods, become infeasible. Virtual testing seems like a promising alternative to traditional methods, especially as part of a scenario-based verification and validation methodology. But in order to transfer the test results of a system from a simulation to the real world, we need to argue the validity of the virtual tests. Our proposed method enables this validity argumentation by comparing the virtual test traces against traces that have sufficiently similar recorded real-world traces. To reduce the amount of required real-world data, the method involves two mechanisms to generalize the validity statement of a single real-world trace to a set of virtual traces. The reduction of required data is showcased in a proof of concept that compares the needed amounts of data with a "naive" validation method and here presented enhancements in an ablation study.

Keywords: Automated Driving, Automated Driving Systems, Computer Simulation, Simulation-based Testing, Validity, Virtual Testing, Virtual Validation

1 Introduction

Automated driving systems (ADSs) [16] are not only thought of as a way to make traveling more comfortable but also as a tool to make it safer [7]. To realize this, it is hence very important to guarantee their safe operation. However, not only the ADSs themselves are highly complex but also the perceived environment which acts as input for such systems. Thus, their careful verification and validation is of high importance. This cannot be done solely with exhaustive (formal) methods as the environment bears infinitely many characteristics, possible interactions and effects [11]. However, when "only" using tests, the number of test kilometers necessary for a statistical evidence of the system's safety amounts to (depending on assumptions/type of accident) several hundreds of millions of kilometers [21] (for comparison: all streets in the US amount to 6.59 million kilometers [2]). In theory, these tests would need to be performed with every newly developed or even slightly modified (e.g. updated) ADS. Hence, this approach is infeasible due to constraints on costs and time [8]. A potential solution to this dilemma is to replace physical components with virtual ones mimicking the behavior of their physical counterparts. If part of the testing of an ADS shall be performed in the simulation, one has to provide plausible evidence that the simulation environment is sufficiently similar to the real world in the required aspects [9]. Otherwise, any argument obtained from the simulation cannot contribute to a safety assessment in reality. This sufficiently realistic behavior is traditionally called model validation: “substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model” [19]. To be able to show this validity, typically, the ADS’ performance in the virtual world is compared to its performance in the real world while executing the same scenario [1]. But since this does not reduce the amount of needed real-world tests, this method might just be helpful during the development phase, where simulations could already be carried out for a faster and cheaper feedback loop.

In this work, we present a snippet based trace validation method that can reduce the required real-world data for virtual ADS verification. Here, a *trace* is a time series of (multidimensional) simulator states $s_l \in S$, where S is the state space of the simulator. Snippet based trace validation helps to decide if a single trace generated by a simulation can be determined "valid". Note that this is different from traditional model validation as we do not determine a priori a model to be valid but only look at single outputs of the simulation platform (a trace). The general strategy is based on a set of validated simulation traces, for which comparable real-world counterparts were gathered. We then generalize the validity statement of a "known" simulation trace to "new" traces, which we have not already seen in reality by arguing carefully over similarities and recombination. Not having to record every slight variation of a scenario in the real world, our method paves a way towards affordable valid virtual testing of ADSs.

To sum up, this paper contributes to answering the following research question: **How can the amount of needed real world data for the validation of simulated traces be decreased?** To answer this question we

– present a method to validate simulation traces based on real world data,
– propose two enhancements via decomposition and recombination of real world data along (1) time and so called (2) validity aspects, and finally
– demonstrate the effectiveness of these enhancements.

The paper is structured as follows: we start with referring to related work in the field of model validation in Section 2. After that, we describe the method and its individual parts in detail in Section 3, which will then be applied in our proof of concept in Section 4. We conclude this paper with a short summary and an outlook on future work in Section 5.

2 Related Work

Validation of computerized models has been a widely recognized problem across multiple disciplines since the beginning of their existence. Even though this paper only looks at single simulation traces and hence not directly talks about model validation, these concepts are related as the here presented method may be used to e.g. create a statistical hypothesis test and thus do model validation.

An overview over the topic of model validation, not specific to any application domain is done by Sargent [17], while Oberkampf et al. present the state of the art in the validation of computational physics models [10]. They conclude that, because erroneous conclusions based on modeling and simulation may have disastrous outcomes, it is important to drastically improve the confidence and understanding in computational simulations. The general challenges to determine the degree of model validity are also relevant to the validity of virtual ADS verification. However, in this context the topic has not mainly been addressed for the simulation of vehicle dynamics or sensors.

Viehof proposes a method for a requirement based validation concept in vehicle dynamics simulation [20]. He proposes different validation level where the first one is *statistical sample validity*. The method presented in this paper can be seen as being on this sample layer. With some modifications, this method was applied to an idealized radar sensor model by Rosenberg et al. [15]. However, they reported several challenges (e.g., find more appropriate metrics). In a follow up paper, Rosenberg at al. present a suggestion for a more appropriate metric for the validation of sensor models [14]. However, the extrapolation from the validated samples is still limited.

Schaermann also proposes a validation method for sensor models [18]. As the approaches before, he relies on re-simulation of real world data. He also addresses different levels for sensor model validation: raw data vs. object level.

Riedmaier et al. present a unified framework and survey for model verification, validation and uncertainty quantification [12]. They applied their framework to a *Lane Keeping Functional Test* and assessed the validity via a minimum distance'to line' [13]. This framework was tailored for statistical validation with a focus on uncertainty, e.g. arising from the input data by Danquah et al. [4, 5]. In their approach, an extrapolation from validation scenarios is also considered and solved by uncertainty learning and prediction which leads to a quite high prediction uncertainty. In contrast to the aforementioned work, where an extrapolation is done via statistical arguments, the here presented method is based on a generalization.

3 Snipped based Trace Validation

Demonstrating the validity of all simulation traces a simulator can generate is hard, especially in the automotive context, where we encounter a highly diverse environment with myriads of possible variations, commonly referred to as the open world problem [3]. This is why we reduce the complexity of the validation problem here by assessing the validity for single simulation traces. Those validated simulation traces can then be used during testing in order to reduce the amount of real world testing. In the following we delineate the basic process, which will be extended subsequently to enable a broader application of the method and, thus, increase its usefulness.

3.1 Naive Validation of Simulation Traces

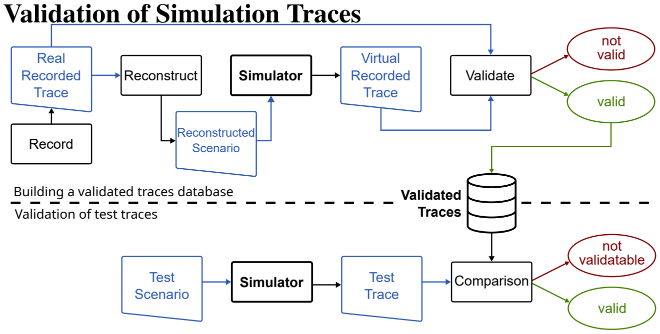

Fig. 1: Naive validation process. Black rectangular boxes represent the process steps, while blue sharp boxes represent artifacts. The process is divided into two phases.

The naive approach of validating a single simulation trace follows a simple idea: If you can find a trace from the real world that is similar enough to the one you have seen in the simulation, you may conclude that it is a valid trace. If we assume that all relevant information can be observed and, that it is sufficient to observe the validity at the sampling rate of the ADS, similar enough real and simulated data indicate that the simulation trace can be used as a replacement of the real-world trace.

This idea leads to a two-phase process: First (Figure 1,upper half), one needs to record data in the real world with a sensor-equipped vehicle (e.g., cameras, radars, lidars, CAN bus data), resulting in a *real recorded trace (RRT)*, which is a time series of all measurements. The sensor-equipped vehicle may be driven by a human or an ADS. The RRT is used to reconstruct simulation-suitable representations of the scenery and the scenario. These files have to describe all relevant properties of the real recorded trace, so that it can be replayed by the simulator in the next step. By gathering similar measurements from the simulation as those obtained in the real-world drive, the replay results in a *virtual recorded trace (VRT)*. In the *Validate* step, the RRT and the VRT are compared against each other. If they are not similar enough, the VRT is considered to be invalid and there must be an error either in the scenario reconstruction or in the simulator setup that should be fixed. In case of sufficient similarity, the VRT is valid and added to the *Validated Traces* database. If one is now interested in testing an ADS using a test scenario in the simulator (Figure 1, lower half), we compare the resulting test trace against the validated traces in the database. If the test trace is not similar enough to an already validated trace, we cannot argue about the validity of the test trace (i.e., it can not be validated at this point), since real-world data might simply be missing. But if a match can be found, the simulative test result is valid. Only comparing the traces is enough, if we assume that the simulator is deterministic and executes the semantic of the scenarios description.

Note that, since the validity of simulation traces is based on the RRT, there exists a coupling between the validity and the modalities of the recording campaign (e.g., sensor setup, vehicle dynamics, ...). When deploying a virtually tested ADS, the physical vehicle has to match those modalities of the recording campaign, otherwise the test results can not be transferred to reality. This restriction stems from the purely data-based nature of our validation process. If adjustments to the physical vehicle are made and one can argue, that those adjustments either have no impact on all measurable properties, or that those adjustments are correspondingly reflected in the simulation model, then this restriction could be relaxed.

3.2 Slicing of Traces by Validity Aspects

In order to accurately reflect the real world, a simulator must include all the relevant aspects of the real world, together with their relations. In the context of ADS, examples for these aspects are the used sensors or the vehicle dynamics. Some of these aspects might work independently from each other, e.g., a radar may be independent from a lidar sensor, due to their different frequencies. We call these independent aspects *validity aspects (VA)*, which can be implemented in the simulation as standalone modules, while only relying on the commonly shared simulation state. Please note, that these aspects have to be chosen *very* carefully (e.g., not all modules in a simulation can automatically be VAs). If there are no independent aspects, then there is just one VA that includes everything. A VA does not have to depend on every variable in the scenario (e.g., objects, daytime, weather conditions) but only on a subset, the so-called *key variables*. For example, a long-range front radar is independent from objects that are not in its field of view, therefore these objects are not part of the key variables of this radar VA (visualized in Figure 2). We refer to such a reduced trace as *pattern*, which now builds the basis for the comparison of a test trace against the validated traces. A trace of length l $\begin{matrix} s_{0,0} & \cdots & s_{0,l} \\ \vdots & \cdots & \vdots \\ s_{n,0} & \cdots & s_{n,l} \end{matrix} \in S$ is split into traces of a VA $\begin{matrix} va^{j}_{0,0} & \cdots & va^{j}_{0,l} \\ \vdots & \cdots & \vdots \\ va^{j}_{m,0} & \cdots & va^{j}_{m,l} \end{matrix}$ and reduced to key variables. This is called a pattern of a VA j $\begin{matrix} k^{va_j}_{0,0} & \cdots & k^{va_j}_{0,l} \\ \vdots & \cdots & \vdots \\ k^{va_j}_{r,0} & \cdots & k^{va_j}_{r,l} \end{matrix}$, where $n \geq m \geq r \in \mathbb{N}$. Since a pattern drops unnecessary variables for this VA, it can now generalize to a lot more traces. And the more variables can be dropped, the better this generalization ability is. However, when identifying key variables for a validity aspect, we recommend to start with all available variables as key variables, and then remove those variables, which are *sure* to be irrelevant for the VA e.g., with a sensitivity analysis. If no error was made, no incorrect generalization can occur.

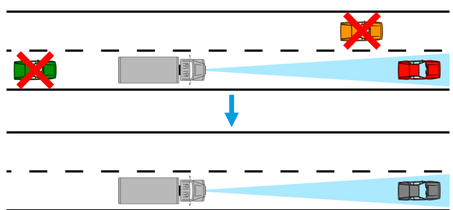

Fig. 2: Visualization of the reduction to key variables for a long-range front radar. Objects outside the field of view and the color of the vehicle can be removed.

When replaying the RRTs, their corresponding VRTs have to be valid w.r.t. all VAs, i.e., the simulator accurately represents the real world in every aspect for each RRT. But when now simulating a test trace and performing its validation, we do not have to find a validated trace in our database that completely matches the new trace, but we can find patterns originating from different validated traces to validate each VA individually. This way, a limited database of validated traces can be used to validate a lot more test traces, reducing the effort for real data acquisition. However, when arguing that these patterns together now form a validated trace great care has to be taken. On an information level there may be emergent behavior that gets lost when only looking at parts. In this case, splitting into validity aspects is prohibited. Moreover, the coupling of the VAs may introduce timing effects that are different in simulation and in reality (eg. FMUs for Co-simulation vs CAN bus in a real vehicle).

Example: Assume we consider two VAs "front camera" and "rear camera". If we recorded two traces in the real world, where 1) the ego vehicle follows another car and 2) the ego vehicle is followed by another car, then for a new test trace where both cases occur at the same time, we can take both existing validated traces to validate the different VAs and therefore the whole test trace.

3.3 Slicing of Traces by Time

When a camera is taking an image, only the light that hits the sensor during its exposure time influences the resulting picture. Analogously, the measurements of active sensors like radar or lidar depend only on the time span between the emission of a ray and its detection. These observations motivate the definition of a maximal time frame – we call it *history* of length τ – in which changes of variables have an effect on a VA at timestep

t. The history of each VA might be influenced by some individual factors, e.g., for the environment VA, physical laws are of major relevance. For the sensor VAs, deriving the length of the history might be as easy as looking up the technical specifications. In general, for each VA one has to examine all the relevant causal chains and their respective lengths. The longest relevant causal chain for a VA determines its minimal history length. After deriving the appropriate histories for all VAs, the patterns for each VA can be time-sliced into even smaller chunks with the length of their histories τ, using a sliding window approach. Since the originating patterns of these snippets were already validated, the resulting snippets are referred to as *validity snippets*.

In order to validate a test trace with these validity snippets, the test trace has to be sliced by VAs first and then by time using the same key variables as they were used during the elicitation of the validity snippets. It is very important, that the sliding window must use a stride so that consecutive snippets of the test trace overlap with each other, since otherwise discontinuities could arise, which should not be considered valid (e.g., teleporting vehicles). A safe decision for the stride of the sliding window is a single time step $\Delta t = \frac{1}{f}$, since we assume that the sampling rate f is sufficient to observe the validity. Using a greater stride must be accompanied by a sound argumentation, that this cannot lead to unreasonable discontinuities.

Having collected all snippets of the test trace, they have to be matched against a validity snippet and only if all these matches were successful, the test trace is considered to be valid. By doing so, it is not necessary to observe a test trace in its entirety in the real world, but just its individual parts.

4 Proof of Concept

To showcase the effectiveness of the different presented techniques (*slicing by VA* and *slicing by time*), we performed an ablation study to quantify the amount of required validated traces to validate a set of test traces. For the sake of simplicity, in this case we only assume the validated traces to be valid without having comparable real world data as we focus on the second phase of the presented method. For this, we used the simulator CARLA 0.9.13 [6] and designed a parameterizable highway scenario, where the ego vehicle is following its lane with a desired target velocity $v_e \in [V_{MIN}, V_{MAX}]$, while adapting to its traffic ahead. The initial position is set to the midpoint of a straight highway, but can be longitudinally varied with an offset $o_e \in [-O_{MAX}, O_{MAX}]$. $n_o \in [N_{MIN}, N_{MAX}]$. Other vehicles with target velocity $v_{o,i} \in [V_{MIN}, V_{MAX}], i = 1 \ldots n_o$ are randomly sampled around the ego vehicle, also adapting their velocities to other traffic, while additionally having the option to perform a lane change, if this is enabled ($lc \in \mathbb{B}$).

Table 1: Overview of parameter ranges for different benchmarks

	N_{MIN}	N_{MAX}	V_{MIN} [km/h]	V_{MAX} [km/h]	O_{MAX} [m]	lc
easy	0	3	80	80	100	0
medium	0	5	80	90	150	1
complex	0	10	60	100	500	1

Based on this parameterizable scenario, we derived three benchmarks *easy*, *medium* and *complex* which differentiate by the used parameter ranges (see Table 1). We sampled 200 and 5000 scenarios for each benchmark as test and validated traces.

We implemented the different validation schemes (*naive*, *slicing by VA*, *slicing by time*, *slicing by VA + time*) and evaluated the overall amount of test traces that can be validated in each benchmark. We used a front-looking and a rear-looking camera as well as the ego vehicle dynamics as validity aspects. The patterns for the VAs of the cameras are reduced to the visible objects at each timestep of the trace. To develop an estimate of how the amount of validatable test traces scales with the amount of validated traces, we performed the evaluation on different database sizes. The results are depicted in Figure 3 and indicate that the different enhancements introduce a great benefit to the amount of validated test traces. The snippet based trace validation is capable of validating 1.7-2.2x more test traces than the naive method. In the *easy* setting, nearly 60% of all test traces can be validated, but the performance drastically reduces for the harder benchmarks, due to the combinatorial complexity of all possible variations.

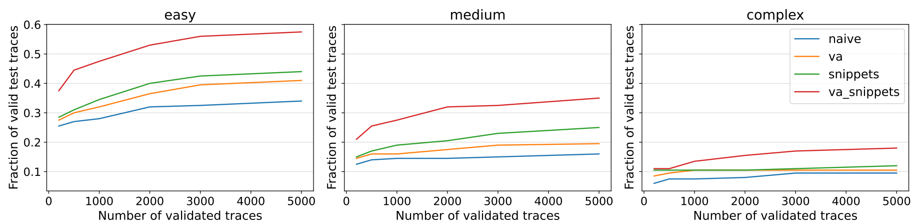

Fig. 3: Results of validating the test traces in the different benchmarks.

5 Conclusion and Future Work

The here presented snippet based trace validation provides evidences for the validity of simulation traces, in order to enable the transfer of virtual testing results to the real world while reducing the amount of needed real world data. It relies on generalizing from traces that were observed in reality and validly replayed in the simulation (the validated traces). Our proof of concept exemplary showed, that our different strategies help to increase the generalization of observed scenarios to multiple similar scenarios. The approach should be applied with great care as wrongfully declared valid traces, have the potential to lead to catastrophic events. This is why research in this area is especially important.

Furthermore, the presented method comes with several limitation. The problems of coupling introduced in Section 3.2 will need further investigation as the introduced timing effects may affect the overall validity. Defining good similarity measures for traces are a further topic of interest. An additional step may be needed to include emergent behavior. Furthermore, our proof of concepts only looks at sensor related validity aspects. The usefulness for other validity aspects needs to be examined further.

In order to benchmark our method, we plan to extend our simulative proof of concept to a broader scale and apply it to actual real-recorded scenarios, thus being able to give estimates on the efficiency of snippet based trace validation in a real-world setting. Here, it will be especially interesting how the combinatorial complexity will affect the performance. To make use of real world data the trustworthiness of data also has to be considered.

Acknowledgments This work has received funding by the German ministries BMWK and BMBF within the projects “KI Wissen”and “ASIMOV”.

References

1. ISO 19364 - Passenger cars — Vehicle dynamic simulation and validation—Steady-state circular driving behaviour. Tech. rep., International Organization for Standardization
2. The world factbook (2019), washington, DC 20505
3. Bendale, A., Boult, T.: Towards open world recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
4. Danquah, B., Riedmaier, S., Meral, Y., Lienkamp, M.: Statistical Validation Framework for Automotive Vehicle Simulations Using Uncertainty Learning **11**(5)
5. Danquah, B., Riedmaier, S., Rühm, J., Kalt, S., Lienkamp, M.: Statistical model verification and validation concept in automotive vehicle design. Procedia CIRP **91**, 261–270 (2020), enhancing design through the 4th Industrial Revolution Thinking
6. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning (2017)
7. ERTRAC Working Group "Connectivity and Automated Driving": Connected automated driving roadmap (2019)
8. Kalra, N., Paddock, S.M.: Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? (2016)
9. Neurohr, C., Westhofen, L., Henning, T., de Graaff, T., Möhlmann, E., Böde, E.: Fundamental Considerations around Scenario-Based Testing for Automated Driving. In: 2020 IEEE Intelligent Vehicles Symposium (IV). pp. 121–127 (2020)
10. Oberkampf, W.L., Trucano, T.G., Hirsch, C.: Verification, validation, and predictive capability in computational engineering and physics. Applied Mechanics Reviews **57**(5) (2004)
11. Poddey, A., Brade, T., Stellet, J.E., Branz, W.: On the validation of complex systems operating in open contexts
12. Riedmaier, S., Danquah, B., Schick, B., Diermeyer, F.: Unified Framework and Survey for Model Verification, Validation and Uncertainty Quantification. Archives of Computational Methods in Engineering p. 2655–2688 (2021)
13. Riedmaier, S., Schneider, J., Danquah, B., Schick, B., Diermeyer, F.: Non-deterministic model validation methodology for simulation-based safety assessment of automated vehicles. Simulation Modelling Practice and Theory **109**, 102274 (2021)
14. Rosenberger, P., Schunk, G., Ikemeyer, F., Duong, Q.T.: Validation of Test Infrastructure - from cause trees to a validated system simulation
15. Rosenberger, P., Wendler, J.T., Holder, M., Linnhoff, C., Berghöfer, Winner, H., Maurer, M.: Towards a generally accepted validation methodologyfor sensor models - challenges, metrics, and first results. In: Graz Symposium Virtual Vehicle (GSVF) (2019)
16. SAE International: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Tech. rep.
17. Sargent, R.G.: Verification and validation of simulation models. In: Proceedings of the 2011 Winter Simulation Conference (2011)
18. Schaermann, A.: Systematische Bedatung und Bewertung umfelderfassender Sensormodelle. Ph.D. thesis
19. Schlesinger, S., Crosbie, R.E., Gagné, R.E., Innis, G.S., Lalwani, C.S., Loch, J., Sylvester, R.J., Wright, R.D., Kheir, N., Bartos, D.: Terminology for model credibility. SIMULATION **32**(3), 103–104 (1979)
20. Viehof, M.: Objektive Qualitätsbewertung von Fahrdynamiksimulationen durch statistische Validierung. Ph.D. thesis
21. Wachenfeld, W., Winner, H.: Die Freigabe des autonomen Fahrens, pp. 439–464. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)